NIST Special Publication 500-205

Computer
Systems
Technology

U.S. DEPARTMENT OF
COMMERCE
Technology Administration
National Institute of
Standards and
Technology

NIST

Guidelines for the Evaluation of Virtual Terminal Implementations

Carol A. Edgar

The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering and performs related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Public Inquiries Desk, 301-975-3058.

Technology Services
- Manufacturing Technology Centers Program
- Standards Services
- Technology Commercialization
- Measurement Services
- Technology Evaluation and Assessment
- Information Services

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]

Chemical Science and Technology Laboratory
- Biotechnology
- Chemical Engineering[1]
- Chemical Kinetics and Thermodynamics
- Inorganic Analytical Research
- Organic Analytical Research
- Process Measurements
- Surface and Microanalysis Science
- Thermophysics[2]

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Molecular Physics
- Radiometric Physics
- Quantum Metrology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Manufacturing Engineering Laboratory
- Precision Engineering
- Automated Production Technology
- Robot Systems
- Factory Automation
- Fabrication Technology

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- Reactor Radiation

Building and Fire Research Laboratory
- Structures
- Building Materials
- Building Environment
- Fire Science and Engineering
- Fire Measurement and Research

Computer Systems Laboratory
- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

Computing and Applied Mathematics Laboratory
- Applied and Computational Mathematics[2]
- Statistical Engineering[2]
- Scientific Computing Environments[2]
- Computer Services[2]
- Computer Systems and Communications[2]
- Information Systems

---

[1] At Boulder, CO 80303.
[2] Some elements at Boulder, CO 80303.

NIST Special Publication 500-205

Guidelines for the Evaluation of Virtual Terminal Implementations

Carol A. Edgar

Systems and Network Architecture Division
Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Partial funding for this document was provided by the Internal Revenue Service

November 1992

U.S. DEPARTMENT OF COMMERCE
Barbara Hackman Franklin, Secretary

TECHNOLOGY ADMINISTRATION
Robert M. White, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) has a unique responsibility for computer systems technology within the Federal government. NIST's Computer Systems Laboratory (CSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. CSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. CSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports CSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-205
Natl. Inst. Stand. Technol. Spec. Publ. 500-205, 47 pages (Nov. 1992)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1992

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

Contents

List of Figures

Disclaimer

Certain commercial products are identified in this report. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

Chapter 1

Introduction

The *Government Open Systems Interconnection Profile* (GOSIP) [11] mandates that Federal agencies requiring remote terminal access capability procure products conforming to the International Organization for Standardization Open Systems Interconnection Basic Class Virtual Terminal Service and Protocol [14, 15]. The Federal government is cooperating with the commercial sector to produce an *Industry/Government Open Systems Specification* (IGOSS) [13] which will be referenced by the next version of the Government Open Systems Interconnection Protocol. The IGOSS will expand the types of terminals which can use the OSI Virtual Terminal protocol.

This document, *Guidelines for the Evaluation of Virtual Terminal Implementations*, advances the goals of the GOSIP by assisting a user in determining which Virtual Terminal (VT) implementation, among several candidates, best meets the functional requirements of that user. This document is one in a series of evaluation guidelines for Open Systems Interconnection (OSI) applications. Evaluation guidelines for Message Handling Systems (MHS) implementations [26] and File Transfer, Access, and Management (FTAM) implementations [18] have preceded this one, and evaluation guidelines for other OSI applications, such as Directory Services, will follow.

The Evaluation Guidelines section of this document is not as extensive as for the MHS and FTAM applications. The differences in VT implementations for the same terminal type are not so great as to warrant a rating algorithm. Still, there are functional issues to be considered in a VT procurement and these issues are described within this document. The current and future availability of VT products is also discussed.

1.1 Scope

These evaluation guidelines apply to implementations which have been produced according to the *International Standard for Basic Class Virtual Terminal Service* [14], the *International Standard for Basic Class Virtual Terminal Protocol* [15], part 14 of Version 5 of the *OSI Implementors Workshop (OIW) Stable Implementors' Agreements for Open Systems Interconnection Protocols (December 1991)* [25], Version 2 of GOSIP [11], and the draft Industry/Government Open Systems Specification [13].

1.2 Overview

The contents of this document are organized as follows. Chapter 1 contains an introduction to the document. Chapter 2 gives a VT overview. Chapter 3 presents a VT tutorial. Virtual Terminal profiles are described in Chapter 4. Chapter 5 describes the IGOSS and the VT functionality contained in the Industry/Government Open System Specification. Chapter 6 specifies the procedure for evaluating VT implementations. Chapter 7 describes the current state of VT and projects the status of future product availability. Appendix A presents the relationship between VT and X Windows. Appendix B defines the abbreviations used in this document, and Appendix C provides a glossary of VT terms. Following the appendices is a list of references.

1.3 Acknowledgments

The National Institute of Standards and Technology (NIST) wishes to acknowledge and thank the vendors who provided VT implementations and documentation to assist this project (Control Data, Data General, Digital, Retix, and 3Com). These implementations facilitated the development of this document.

The NIST also wishes to thank Network General and Novell for providing Protocol Analyzers, 3Com for providing OSI lower layer routers, Interactive Systems for providing UNIX operating systems, and Cyndi Jung of 3Com for assisting with the VT tutorial.

Chapter 2

Virtual Terminal Overview

This chapter discusses the need for a virtual terminal service and explains how the Virtual Terminal application meets that need.

2.1 The Need for a Virtual Terminal Service

With increasing frequency, the resources that computer terminal users need to access are located on remote computer systems. This increase in distributed operations on computer networks has accelerated because of the proliferation of local area networks. Computer terminal users need to access application programs resident on local or remote systems and application programs need to communicate with terminals in a way that is independent of the design and characteristics of specific terminal models.

The OSI Virtual Terminal application provides a solution to this problem. The VT application provides mechanisms to effectively insulate application processes from the specific characteristics of the terminals with which they communicate. This allows terminals to access applications running on a wide variety of systems, and vice versa, regardless of the supplier of the terminal or host system. Virtual Terminal ushers in a new era where people are no longer limited to buying terminals and computers from a single supplier. Hence, VT facilitates intercommunication in a multivendor environment.

2.2 Achieving the Virtual Terminal Goal

As its name implies, the VT service is provided by defining a virtual terminal which is a generic representation of all the operations possible on a user's terminal. This generic representation is then mapped onto the specific characteristics of the terminal.

Before interactive communication begins, the VT application serving the terminal user must establish an association with the VT application serving the application process. In establishing this association, agreement must be reached not only about the data which can be transferred (e.g., the character set) but also about information relating to the form in which data is displayed on the terminal (e.g., foreground color, background color, character font) and the terminal control functions (e.g., echoing, mapping of carriage return) which can be manipulated.

Once agreement is reached, the terminal can communicate with the application program via the agreed virtual terminal. Similarly, the application program need not be concerned with characteristics and control functions specific to the terminal with which it wants to communicate. Rather, the application program only needs to be concerned with the virtual terminal defined for that particular association, safe in the knowledge that the real terminal can translate information on the virtual terminal into its own terminal-specific control functions and characteristics.

For a detailed explanation of the mechanisms used to provide the VT service, see Chapter 3.

Chapter 3

Virtual Terminal Tutorial

The Virtual Terminal service provides for terminal to application communications in the OSI environment. The service ranges from very simple interactions to extremely complex ones. As with most OSI applications, flexibility and scalability are valued over simplicity, and the Virtual Terminal Protocol (VTP) has flexibility. In order to achieve this flexibility, the standard provides a model, a set of tools and a rule book for creating an interactive Virtual Terminal Environment.

Virtual Terminal is a service of the OSI Application layer and uses the services of the Application Control Service Element (ACSE), Presentation layer, the Session layer and a suitable T-Profile stack as defined in ISO/IEC TR 10000-2, the Information Technology - Framework and Taxonomy of International Standardized Profiles - Part 2: Taxonomy [16]. The VT services and protocol are specified by the International Organization for Standardization (ISO) in ISO 9040/9041 [14, 15]. Figure 3.1 presents a pictorial view of the OSI Reference Model and the mapping for the VTP Basic Reference Model.

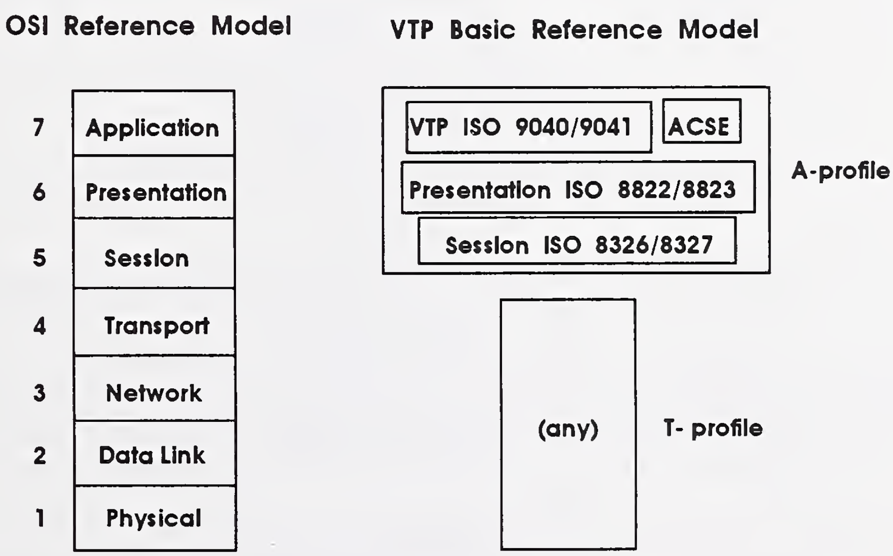

Figure 3.1: The Mapping of the Virtual Terminal Protocol to the Open Systems Interconnection Reference Model

Understanding the information presented in this chapter is necessary background to appreciate the material covered in Chapters 4, 5, and 6.

3.1 Definitions

This section presents definitions key to the understanding of Virtual Terminal. For a more complete list of acronyms and definitions see Appendices B and C, respectively.

3.1.1 The Virtual Terminal

In the past, users accessed an application from terminals designed to interact directly with that application. Today, with the need for multivendor networking and the introduction of open systems, users access applications with a variety of terminals. Regardless of the make or model of a terminal, users need to access a variety of application programs on local and remote computer systems. The OSI VT protocol attempts to solve this problem by defining a *virtual terminal.*

The virtual terminal models the operations which may be performed on a real terminal, and which are understood by both terminal and application. Users communicate by writing to and reading from the virtual terminal. The virtual terminal is translated by each side, using a local mapping, into its own terminal-specific functions and characteristics. Figure 3.2 illustrates the virtual terminal model.

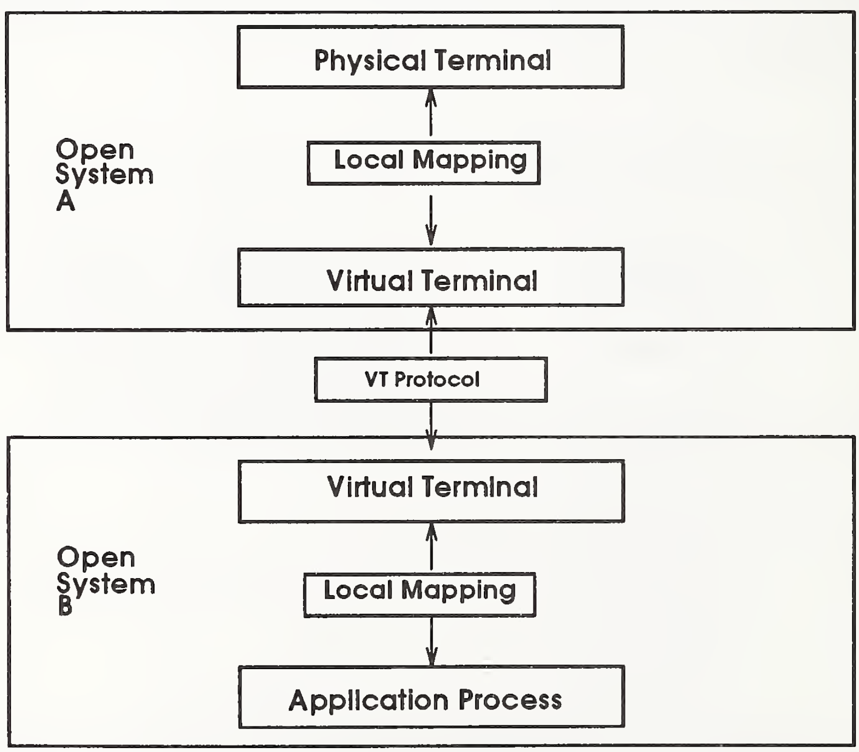

Figure 3.2: Virtual Terminal Model

3.1.2 Virtual Terminal Environment

The agreement of the context within which the virtual terminal will live is known as the Virtual Terminal Environment or VTE. The process which serves the user who initiated the communication flow is called the *initiator*. The *responder* is the process on the remote systems which provides the services required by the initiator. The link established between the initiator and the responder is known as a VT association.

The two users communicating may be identified by one of the following scenarios:

- two application processes
- two terminals back to back or, most common,
- a terminal initiating a VT association with a remote process, as seen in figure 3.3.

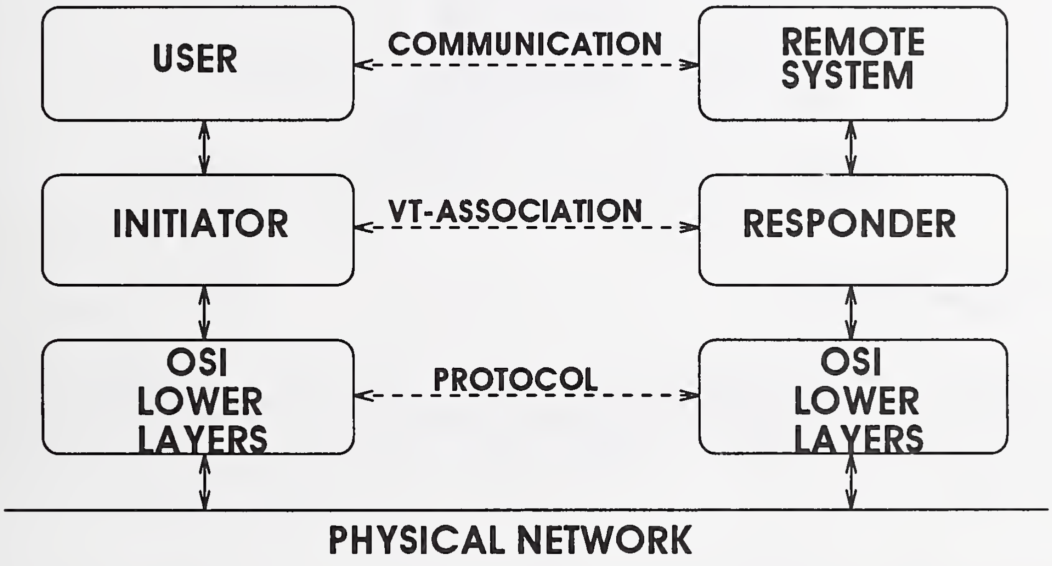

Figure 3.3: Virtual Terminal Communication

3.1.3 Conceptual Communications Area

The VTE contains the abstract or conceptual objects of the VT model. These objects are organized within the Conceptual Communications Area (CCA). The most prominent objects in the CCA are the display objects, control objects, and device objects. These objects are described more fully in the following sections and the relationship between the objects is illustrated in figure 3.4.

3.2 Virtual Terminal Objects

The next few sections describe each of the object types which may be found in the Conceptual Communications Area.

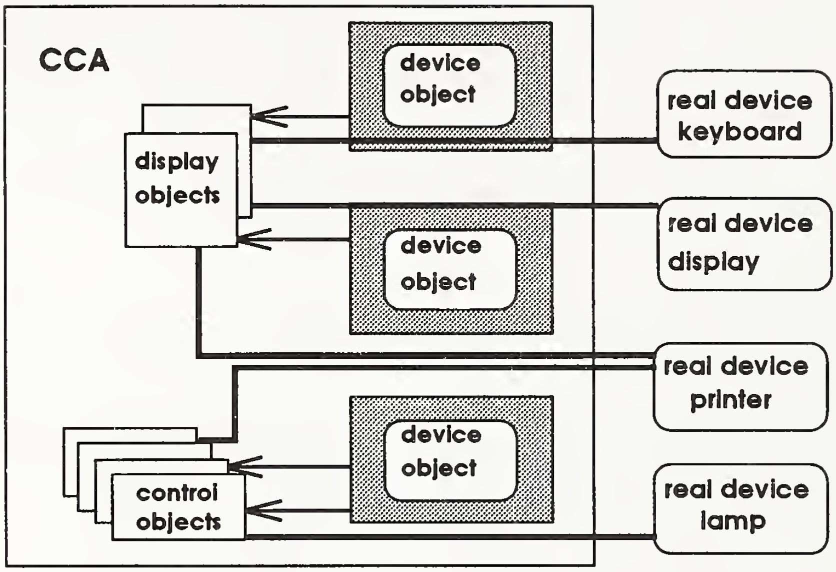

Figure 3.4: Conceptual Communications Area Components

3.2.1 Display Objects

A display object is an abstract object, used to model devices such as the screen or keyboard. A display object contains the graphic information associated with a real device. A display object allows real events which happen on a real terminal to be represented on the virtual terminal as abstract events.

Display objects are defined and may be visualized as one, two, or three-dimensional arrays of character-box elements. Each element of an array may contain a single character, i.e., primary attribute plus a variety of secondary attributes. The secondary attributes describe the element's characteristics. Possible secondary attributes are color (foreground and background), emphasis, and repertoire. The repertoire may have a font type as a subordinate parameter. Figure 3.5 illustrates a display object.

The VT service provides a set of operations that may be performed on a display object. These operations are divided into two types: addressing operations and update operations. The parameters of the display object control the kinds of operations which can be performed on it. The dimension parameters control the addressing operations. For example, it may be permitted to move the display pointer backwards and forward in a given X-array (line of characters) or Y-array (lines on a screen or a page), but the only way to move along the Z-dimension (between pages) is via implicit addressing from the last array element of the previous Y-array. The display object has capability parameters controlling the use of certain operations. For example, the erase capability parameter must have the value "yes" in order for the ERASE operation to be used.

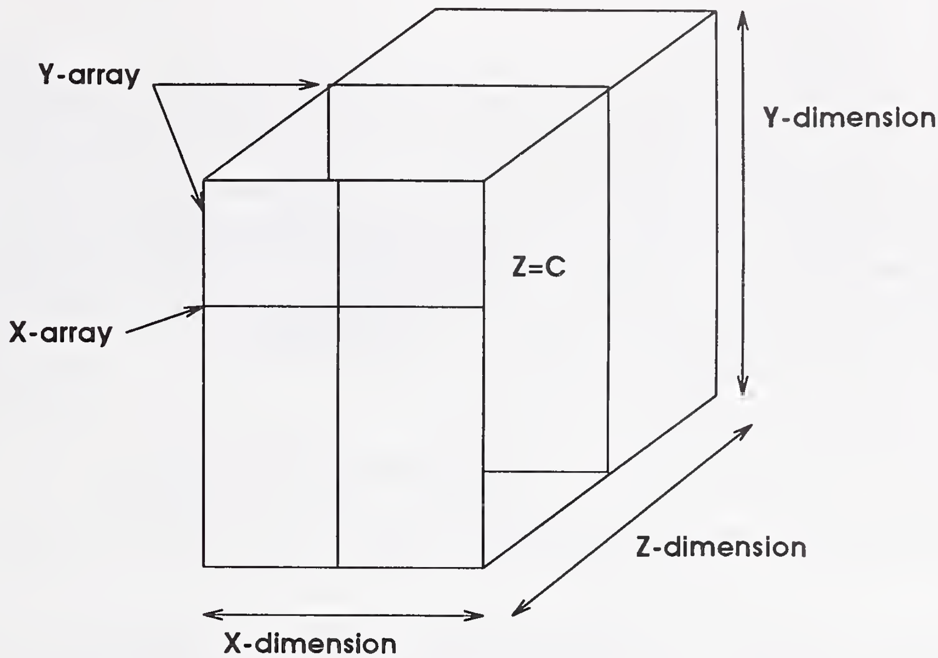

Figure 3.5: Display Object

Below is the complete list of parameters associated with a display object:

- display object name
- dimensions
- erase capability
- attributes
- display object access
- block definition capability
- item field definition capability

As mentioned before, the dimension parameters control addressing operations. The dimension parameters consist of the upper bound of the dimension, the size of the window, a constraint on forward relative addressing, and a constraint on backward relative addressing. The erase capability parameter controls the use of the ERASE operation. Attributes describe the element's characteristics, e.g., emphasis, color and repertoire. Update operations on a display object are restricted to the following operations: text, repeat text, attribute, or erase. The display object access parameter defines the access rights of a display object. All update operations are subject to the access right for the display object. (Please refer to Sec. 3.4.6, for information on VT access rights.)

Structure may be imposed on display objects by defining blocks and fields. A block is a simple subdivision of the display to aid in addressing. The block definition capability parameter records values necessary for a block, such as the block boundary. A field is a nonoverlapping area of a display object used for the validation of human user entry. The field definition capability allows for the definition of a field with parameters like "max-fields", "max-field-elements" and "access-outside-fields."

3.2.2 Control Objects

The control objects of the VTE are much less rigidly defined than display objects. Like display objects, control objects are abstract objects which can be used to model aspects of real terminals such as bells or light-emitting diodes (LEDs), or to handle control information. Control objects can also be used for modeling the exchange of any unstructured information of a single type. The unstructured information is not limited to information of a control nature, although this is the primary application. Because control information is sometimes very specific to applications and terminals, the semantics of the control information are not a part of the VT standard. The semantics of the information content of a control object are:

1. defined in a VT profile,

2. defined as part of a registered control object, or

3. made known to VT users by means outside the scope of the VT standard.

A precise definition of control information semantics is required for VT to operate correctly. Since the VT standard does not provide the mechanisms, control information semantics must be specified in profile definitions.

3.2.3 Device Objects

Device objects are used to represent real devices such as a keyboard, a screen, a printer, or a lightpen. Device objects are intended to assist in mapping display objects to real devices. For example, assume that a display object supports green, amber, blue, and black as foreground colors but the actual device supports only green, amber, blue, and red as foreground colors. The device object will contain the mapping from "black" to "red." This mapping capability provided by the device object allows the same display object to support more than one device type.

3.3 Virtual Terminal Communication Modes

The VT protocol provides for two modes of operation: asynchronous, or A-Mode, and synchronous, or S-Mode. A-Mode operation provides full-duplex communication by using two monologues (in opposing directions) to provide update access to two display objects. Figure 3.6 shows the communication flow

for A-Mode. Update access to each display object is controlled by a different, nonreassignable access right. Each display object has an access right assigned for the duration of the VT association. These access rights give one user the sole right to update that particular display object. In A-Mode there can be zero or more control objects. Each control object has an access right, which is assigned for the duration of the VT association.

S-Mode operation provides half-duplex communication using one, two-way-alternate dialogue supporting one display object. Figure 3.7 shows the communication flow for S-Mode. Update access to the display object is controlled by a single reassignable access right. Access rights to control objects also conform to the half-duplex mode of operation and utilize the same reassignable access right.

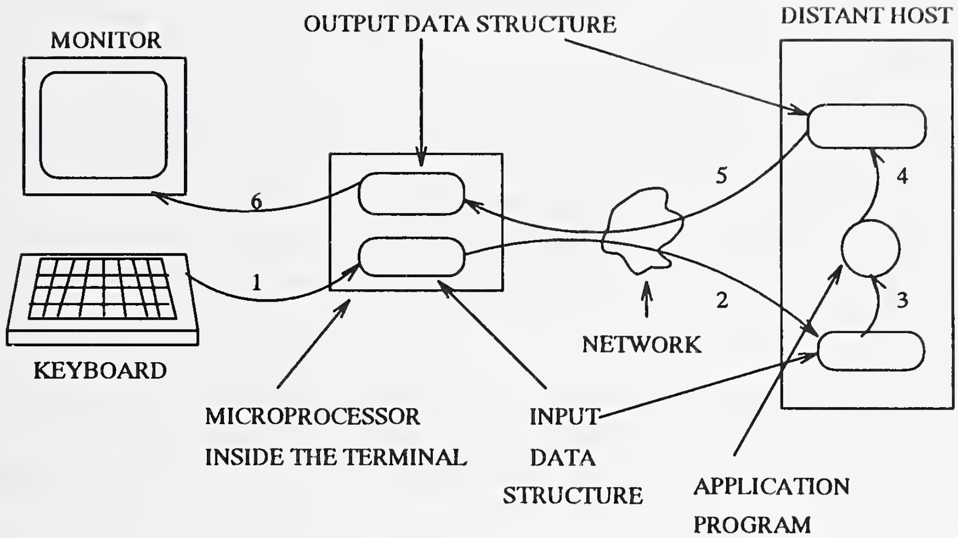

Figure 3.6: Asynchronous Mode Communication Flow

3.4 Virtual Terminal Service Facilities

The Virtual Terminal Service standard [14] partitions the services provided by the VT application into communication facilities. This section explains each communication facility and briefly describes the protocol mechanisms required to implement that facility.

3.4.1 Establishment

The VT Establishment facility provides a service that establishes the Virtual Terminal Environment (VTE) in which the VT data transfer will take place. The initiating VT user proposes a VTE which can be accepted, rejected, or negotiated by the responding VT user.

Negotiation during establishment includes selecting a profile and communication mode. A *profile* is a subset of features of the VT protocol that must be supported for a given connection. A profile has a name and is defined by a set of parameter values. A profile may be incomplete; that is, some parameters are unspecified. These unspecified parameters are numbered. During the establishment of the VT association, the initiator and responder must agree to values for the unspecified parameters. The association requestor or initiator specifies the profile name and offers values for the unspecified parameters. The offered values for the unspecified parameters may be specific values, lists or ranges of values. The association acceptor or responder then chooses the actual values from the offer. For example, the requestor may offer a line length range of 80 to 132 characters. The acceptor must pick a length in that range.

The communication mode for the association must be established. The communication mode is implied or designated by the profile. If no profile is specified, the default profile for the communication mode selected (A-Mode or S-Mode) is assumed. The communication mode selection is necessary to establish access rights and dialogue control for the VT association. The access rights are used to control access to specific VT objects, such as control objects. The access rights also aid in dialogue control, which provides communication integrity by detecting and resolving collisions of service primitives.

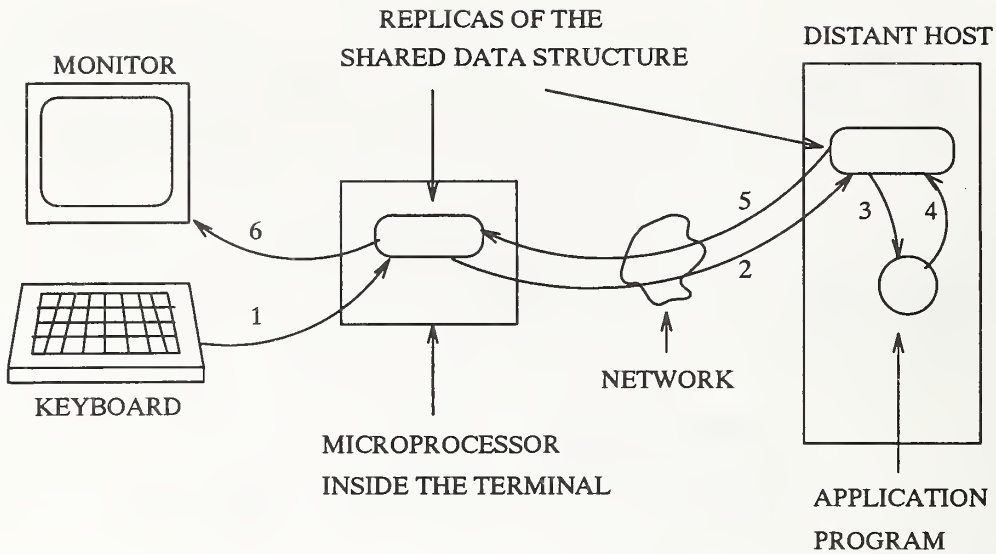

Figure 3.7: Synchronous Mode Communication Flow

In S-Mode, the access rights are reassignable, therefore the initial owner of the access right or token must be determined. The VT standard defines the following mechanism to determine the initial owner of the token. The association requestor selects one of three possible options:

1. the requestor or initiator side,
2. the acceptor side, or
3. the acceptor chooses.

If the third option is selected, the acceptor chooses the initial owner of the token and includes the value in the association response.

In A-Mode the access rights are not reassignable; each VT user is assigned an access right. Since each VT user is in possession of an access right, both are able to issue requests for certain VT services. This results in collisions. For example, the two VT users may simultaneously send a request to start negotiation. To handle collisions, the VT standard designates one user as the collision winner, and provides rules for collision detection and resolution.

The establishment facility is provided by the "VT-ASSOCIATE" service primitive, which is a confirmed service primitive; that is the responding VT user replies to the proposed VTE with *success, failure, or success with warning.* In the latter case, negotiation may be used to further define the VTE or the association may be terminated.

3.4.2 Data Transfer

A second VT facility is known as Data Transfer. The user specifies, by name, the object to receive the data, followed by the data itself. Objects receiving data are either control objects or display

objects. Data transfer can thus be separated into two facilities: control data transfer and display data transfer.

Control data transfer is the transfer of data to a control object. This is more commonly referred to as "updating the contents of the control object." Control data transfer is simple and straightforward. A control object can be thought of as a data structure or variable, which can be a bit string, an integer, a symbolic value, plus other data types, or may even be a sequence of these values. For a boolean string, one boolean may be updated at a time. For all other categories, the content of the object is overwritten.

Display data transfer, involving the updating of the contents of a display object, is more complicated than control data transfer. There are four possible update operations which may be performed on a display object. These operations may put characters into cells, change the attribute associated with a set of cells, move the display pointer, or erase the contents of a set of cells. The following paragraphs explain the details of the display object update operations.

The text operation puts characters into cells starting from the current display pointer position. This causes an implicit change in the position of the display pointer, incrementing it to the next cell on the line. When the last cell in the line has been updated, the display pointer moves one position past the end of the line. The position of the display pointer does not move until an explicit addressing operation is done.

The attribute operation assigns attributes to the cells. The user specifies the attribute, its value, and the extent of the display object that receives this attribute assignment. Possible extents are:

global - all cells in the display object,

forward - all cells from the current display pointer position to a specified cell,

backward - all cells from the current display pointer position back to a specified cell,

explicit range - all cells between two specified cells, or

modal - all cells that have text operations on them in the future.

Explicit addressing operations change the position of the display pointer. The availability of each operation depends on the addressing constraints on the display object.

home - move the display pointer to the first character of the first line of the first page.

next line - move the display pointer to the first character of the next line.

previous line - move the display pointer to the first character of the previous line.

next page - move the display pointer to the first character in the first line of the next page.

previous page - move the display pointer to the first character in the first line of the previous page.

relative - move the display pointer forward or backward the number of characters, lines, and pages specified.

absolute - move the display pointer to the character, line and page specified.

Erase operations may commence from the current position of the display pointer backward or forward to a specified position. In other words, an erase operation may erase a whole line, page, or book (many pages). The user may specify whether the erase operation should erase only the contents of the cell or the assigned attributes. The availability of erase operations is specified in the display object parameters and is dependent upon the number of dimensions defined.

The VT Data Transfer service is provided by the unconfirmed "VT-DATA" service primitive. The parameters include the VT object to be updated, and, optionally, a request to have the data echoed.

3.4.3 Termination

A third VT facility is the VT Termination facility. There are three methods of terminating a VT association. The user can request an orderly termination with no loss of data. In this case, the request may be accepted or rejected by the other user. The user can abort the association. This causes an immediate termination with possible loss of data. This request may not be refused by the peer user. The service provider may unilaterally terminate the service and notify the user that the service has been terminated at a lower layer.

The "VT-RELEASE" service primitive provides the orderly termination service. One VT user requests a service termination; the responding VT user may optionally provide a reason if the service request is rejected. A user aborts the association by sending a "VT-U-ABORT" request which optionally contains the reason for the abort; this request must be confirmed by the peer VT user. "VT-P-ABORT" is an indication-only service primitive that notifies the user that the association has been terminated by the service provider. The service primitive can optionally contain the reason for the termination.

3.4.4 Negotiation

The Negotiation facility provides services which make it possible for peer VT users to select, modify, or replace the current Virtual Terminal Environment. Negotiation is available as part of the establishment facility. An initial VTE is established during association establishment using the profile-based negotiation embedded in the "VT-ASSOCIATE" service. This initial VTE may or may not be a complete or full Virtual Terminal Environment.

A profile is a set of parameters that specify the capabilities and constraints of a terminal. If the user does not specify an initial VT profile during association establishment, a default profile appropriate for the mode of operation is used. The default VTE or the user specified VTE may be modified or replaced, depending on the type of negotiation facilities available. The following paragraphs describe each form of negotiation.

3.4.4.1 Switch Profile Negotiation

Switch profile negotiation, or profile switching, is the ability to change to a different profile without terminating an association. Profile switching avoids wasting valuable networking resources. A user, using a form of VT which does not support profile switching, who wants to use a different profile must discontinue the current VT association and then set up a new association under the new profile.

Switch profile negotiation allows the user to establish an association in a profile which does not provide grounds for rejection (like the default A-Mode Profile) and then negotiate another profile, like the Generalized Telnet Profile. Frequently, a user's needs will change during a VT session. The user may begin needing the simple line-by-line transmission provided by the default A-Mode Profile, but then wish to use a full-screen editor, which requires the services of a profile like the Transparent Profile.

The Switch profile negotiation is provided by a single confirmed service primitive, "VT-SWITCH-PROFILE". The proposed VT profile name is a mandatory service parameter. This profile can be modified by one or more arguments contained in the service primitive. The request can be accepted or rejected by the peer VT user or by the service provider. If the request is rejected, the reason can be optionally included in the response. The *OSE Implementors' Workshop Stable Implementation Agreements* [25] contain no specific agreements on switch profile negotiation, except to identify it as optional.

3.4.4.2 Multiple Interaction Negotiation

Multiple Interaction Negotiation (MIN) allows for the negotiation of a set of VTE parameters comprising a full Virtual Terminal Environment. Unlike Switch Profile Negotiation, MIN occurs in a series of steps. Negotiation normally continues until a full VTE has been defined, however, either VT user may terminate the negotiation at any time. When a full VTE has been defined, the VT user must then decide whether to adopt the new VTE for use in the current VT association.

Because the complexity of MIN is not justified by an increase in user service, there are no current plans to develop implementation agreements for MIN, and no products which implement MIN exist today. The reader is referred to the VT Service Specification [14] for further details on how this service is provided.

3.4.4.3 No Negotiation

If no negotiation capability is provided, the only way to change the VTE once it has been established and agreed is to terminate the current VT association and establish a new VT association with a new Virtual Terminal Environment. The absence of a negotiation capability supplied by the negotiation facility does not impact or limit the normal option negotiation available in certain profiles after the VTE has been established, e.g., the Generalized Telnet Profile.

3.4.5 Delivery Control

The delivery control service facility allows a user that initiates updates to exercise control over the time at which updates are delivered or made available to the peer VT user. Delivery control can

take one of two forms:

1. The VT user can indicate a point in the data stream at which all previous updates should be delivered to the peer VT user. Updates could have been delivered at any time prior to this point.

2. The VT user can prevent data delivery to the peer VT user until input is completed and checked for correctness.

The first type of delivery control is called simple delivery control. It may be used, for example, to insure data delivery after a full line of text has been input. The second type of delivery control is called quarantine delivery control. It may be used by a page-mode terminal to allow editing before data is delivered to the remote application. A side effect of quarantine delivery is "net effecting." This allows for delivery of the final result, or net effect, of a series of updates.

Only one type of delivery control may be in effect on a VT association at one time. The type of delivery control that is used, if any, is governed by the delivery control parameter in the Virtual Terminal Environment. The "VT-DELIVER" service primitive is used to request the delivery control service. If an acknowledgment of the delivery is requested, it is provided by the "VT-ACK-RECEIPT" service primitive.

3.4.6 Access Right Management

The Access Right Management facility provides services which enable the VT users to control the ownership of access rights which mediate the use of other facilities. An access right is an abstract object representing permission to carry out some set of operations. Some operations of the VT service are subject to access control and may only be performed when an appropriate access right is owned. An access rule is a property of a control or display object which limits access to that object to VT users owning the specified access right.

In asynchronous mode, there are two display objects. (See fig. 3.6.) Each user has write access to one display object and that access right never changes during the VT association. Write access to control objects that are subject to access control may be assigned to either user and that access right never changes during the VT association.

In synchronous mode, there is one display object. (See fig. 3.7.) The initial permission to write to the display object is agreed when the VT association is established and permission can be transferred from one VT user to the other during the association. The VT user that has write access to the display object also has write access to those control objects that are subject to access control.

The service primitives that transfer access rights between VT users are "VT-GIVE-TOKENS" and "VT-REQUEST-TOKENS." "VT-REQUEST-TOKENS" requests a transfer of ownership of the reassignable access right; "VT-GIVE-TOKENS" passes ownership of the access right.

3.4.7 Interrupt

The final VT service facility is the Interrupt facility. The interrupt facility allows a VT user to interrupt a previously initiated sequence of updates to display and control objects, discard all updates currently being exchanged in either direction, and resume exchanging updates after the VT providers have resynchronized their activities. The "VT-BREAK" service primitive interrupts the activities of two VT users and discards all previously initiated object updates which have not been processed.

3.5 Repertoires

Communication between computers – like any form of communication – requires an agreement on the meaning of the symbols used. For example, most computer systems use the American Standard Code for Information Interchange (ASCII) which contains a means of representing numeric, alphabetic, graphics and control characters.

The ASCII is an example of a repertoire. A repertoire may also be a subset of ASCII or any other widely accepted convention for information representation. A repertoire may be created by collating a number of subsets. The resulting repertoire can be given a unique name. In this way it is possible to communicate the identity of the repertoire in use to the other side of the dialogue.

The OIW Implementors' Agreements [25] require support for the 7-bit U.S. ASCII as well as the International Reference Version (IRV) of ISO-646 graphic repertoires for all VT profile implementations.

Chapter 4

Virtual Terminal Profiles

The VT protocol provides a number of features, many of which are optional. Implementations of the protocol vary in the features they support. The VT standard and implementors' groups have defined *profiles* to support today's terminals and applications (see fig. 4.1). A *profile* is a named subset of features of the VT protocol. A profile definition determines the types of applications you can run on a remote host, as well as the terminal types which can be used.

A-MODE PROFILES	S-MODE PROFILES
A-Mode Default	S-Mode Default
Generalized Telnet	Forms
CCITT X.3 PAD	Paged
Transparent	

Figure 4.1: Virtual Terminal Profiles

Similar profiles that are identified by several regions of the world are harmonized to produce one specification which can be used internationally. This process is known as the International Standardized Profile process. International Standardized Profiles (ISPs) have been developed for the Forms and Paged profiles. International Standardized Profiles for other profiles may be developed in the future. The OIW VT Special Interest Group (SIG) plans to submit and advance the Generalized Telnet Profile through the ISP process.

Profiles are the normal starting point in establishing a VT association and the VT standard provides default profiles for asynchronous and synchronous mode. Using profiles to establish a VT association allows the initiator to rapidly convey the user's needs to the responder. Some profiles allow the negotiation of arguments which supply values to VTE parameters. Profiles, therefore, make it possible to establish a VT association quickly and efficiently, while supporting some degree of fine-tuning.

A profile must be selected at the beginning of an association. If the Switch Profile option has been initially agreed by initiator and responder, a new profile may be invoked during an association. To establish agreement on the use of a new profile, the initiator sends the name of the new profile (plus any required arguments). The responder either accepts the proposed switch to the new profile or indicates that the existing profile should continue to be used.

The VT standard defines two default profiles, one for each communication mode. These profiles are called the A-Mode Default Profile and the S-Mode Default Profile. The following sections describe the two default profiles plus the Generalized Telnet, X.3, Transparent, Forms, and Paged profiles. Each section is devoted to one of the profiles and includes a brief description of the profile, plus gives a list of the characteristics over which the user has control. Each section contains an example of the applications or terminals which are applicable to the profile.

4.1 A-Mode Default Profile

The A-Mode Default Profile is mandatory for conformance to the base standards for all OSI VT products which use A-Mode. Any VT user proposing an association with this profile to an OSI VT host supporting A-Mode cannot be rejected on the grounds that the A-Mode Default Profile is not supported.

The A-Mode Default Profile provides a scrolled display of lines of up to 80 characters with no paging. The characters used are graphics characters of International Reference Version ISO 646. The emphasis, color, and font are fixed. Usually, with VT, a user must choose between local or remote echoing. By default, local echoing is not in effect with the A-Mode Default Profile. Most vendors, however, perform a control object update after the association has been established to allow local echoing.

A typical use of this profile is for applications which require simple conversational operation with a two-way-simultaneous (i.e., full-duplex) dialogue. The A-Mode Default Profile may readily be mapped onto conventional teletypewriter devices and teletype-compatible video devices having full-duplex capability. The two display objects usually represent a display or printing device and the keyboard.

4.2 Generalized Telnet Profile

The Generalized Telnet Profile supports services equivalent to those provided by Transmission Control Protocol/Internet Protocol (TCP/IP) TELNET. The Generalized Telnet Profile's network virtual terminal is an asynchronous terminal supporting 7-bit ASCII, and allowing for control capabilities similar to the common facilities available on TTY-type terminals. The Generalized Telnet Profile supports three basic services – data transfer, commands, and option negotiation. Data transfer consists of a sequence of bytes of user or application data which is sent as display object updates.

The Generalized Telnet Profile supports numerous commands such as Status, Binary Transmission, and Echo. These commands are used to create and manage interactive terminal sessions. Commands are sent as control object updates.

Option negotiation allows implementations to agree on enhanced Telnet services for the connection. A user starts negotiation by sending a request for an option. The user receives a response which accepts or rejects the option. Successful negotiation requires that each side begin using the negotiated option. An option can be terminated or renegotiated by the user sending a request to end the option. The user receives a response which accepts or rejects the termination of the option.

The Generalized Telnet Profile was developed to facilitate the migration from the TCP/IP TELNET environment to the OSI VT environment and to support the development of gateways between networks using TCP/IP TELNET and networks using OSI VT. The Generalized Telnet Profile supplies the user with a familiar interface and a set of TELNET-like characters. Below is a list of the Generalized Telnet Profile features:

- the ability to send TELNET commands across the network
- the facility to control certain local terminal characteristics, such as echoing
- the capability of raw binary transmission

Many users access UNIX or Portable Operating System Interface (POSIX) style systems where the native terminal support is TELNET. The Generalized Telnet Profile has been specifically designed to emulate TELNET in this environment.

4.3 X.3 Profile

Triple-X (X.3, X.28, X.29) is a group of standards which define network communications using a Packet Assembler/Disassembler (PAD) meeting the Consultative Committee on International Telephony and Telegraph (CCITT) X.3/X.28/X.29 recommendations. A PAD interfaces between a start/stop terminal and a packet switched network. A PAD provides a "black box" between the terminal and the network, using asynchronous mode to talk to the terminal and communicating in the X.29 protocol with the network. In the CCITT recommendations:

- X.3 defines the PAD parameters,
- X.28 defines the interface between the terminal and PAD, and
- X.29 defines the PAD to computer interface.

Like the Generalized Telnet Profile , the A-Mode X.3 Profile provides a migration path between the Triple-X standards and the OSI Virtual Terminal.

The X.3 Profile, in addition to being developed to facilitate the migration to OSI VT, was developed to help support the development of gateways to OSI VT networks. The X.3 Profile user accesses applications via a Packet Assembler/Disassembler. The X.3 Profile supplies the X.3 Profile user with a familiar interface and a set of X.3-like characteristics.

A major attribute of the X.3 Profile is the ability to set local terminal characteristics. For example, the user can determine very precisely the conditions which cause data to be forwarded to the remote host.

4.4 Transparent Profile

The Transparent Profile allows for the exchange of an uninterpreted sequence of characters. The profile is considered transparent in that it dispenses with the semantics required to provide a model of what is displayed on the remote terminal. Data is transmitted in the form of octet strings with no complex structuring of the data store which models the terminal display. The octets may have any value between 0 and 255.

The Transparent Profile supports applications that wish to control terminals directly through the use of embedded control characters and escape sequences. The profile circumvents the benefits of the virtual terminal profile by not mapping between different vendors' terminal types. Since, the Transparent Profile does not do any mapping between local and virtual terminals, the application must be able to work with the terminal as if it were locally connected.

The profile defines a default character set, the "Virtual Terminal Transparent Set," which is registered under ISO 2375 with register value 125. The profile supports an optional argument, repertoire-assignment. Implementations supporting this optional argument enable the negotiation and selection of an alternate repertoire. The OIW Implementors' Agreements [25] require support of the default character set and strongly recommend that the profile argument, repertoire assignment not be used.

The Transparent Profile has several uses. A few of these uses are listed below:

- control of screen editor
- Videotex
- movement of encrypted data

4.5 S-Mode Default Profile

The S-Mode Default Profile is mandatory for conformance to the base standards for all OSI VT products which use S-Mode. Any VT user proposing an association with this profile to an OSI VT host supporting S-Mode cannot be rejected on the grounds that the S-Mode Default Profile is not supported.

A typical use of the S-Mode Default Profile is for applications requiring simple two-way-alternate (i.e., half-duplex) conversational operation. It is readily mappable onto conventional teletypewriter devices and teletype-compatible video devices.

4.6 Forms Profile

The Forms Profile is an S-Mode profile supporting the use of forms-based, field-oriented, data-entry applications between a terminal and a host system. The profile provides facilities for

- defining and using screen forms,
- defining field validation and field entry rules, and
- controlling and validating field entry.

The Forms Profile also contains capabilities commonly required by other nonforms applications, such as word processors and text editors. The Forms Profile includes support of an optional terminal-end locally attached printer.

During typical transactions, a form is presented on the screen which the user fills in with guidance from the application. A good example of this is when information such as your name, address and telephone number is entered directly into a customer database by the counter personnel when you purchase products in a department store.

The Forms Profile is applicable to a wide range of applications utilizing screens. A typical application requires a complex field structure. The profile allows for the complex field structure and in addition, supports an unformatted paged operation where no fields are defined or where a single field covers the entire screen. It is also possible for the Forms Profile to simulate scrolling.

The Forms Profile has a single display object and 31 arguments, some of which may occur multiple times, yielding lists of values. These arguments supply such things as repertoire, font, color, emphasis, and items relating to fields. Two very important control objects in use in the Forms Profile are the Field Entry Instruction Control Object (FEICO) and the Field Entry Pilot Control Object (FEPCO). These control objects along with the other Forms control objects provide the mechanisms special to applications using fields.

Many of the field entry validation capabilities of the Forms Profile require character-at-a-time interaction with a validation process resident in the terminal's VT system. The implication for the existing terminal population is that the Forms Profile is only suitable for a character-mode terminal base. The Forms Profile **IS NOT** suitable for use in conjunction with traditional block-mode terminals.

4.7 Paged Profile

The Paged Profile is an S-Mode profile reflecting the functionality of existing block-mode terminals, in particular the IBM 3270 style. The profile contains a subset of the functionality of the Forms Profile and represents a natural step in the migration to the Forms Profile.

The arguments of the Paged Profile supply such things as color, emphasis, repertoires, font, and fields. The Paged Profile allows for an optional monochrome or color printing device. As the name implies, the Paged Profile is used in applications where data is displayed on a whole screen or paged basis. The Paged Profile also supports applications using fields.

Chapter 5

Industry/Government Open Systems Specification Requirements

The Federal government and three different communities in the commercial sector (Manufacturing Automation Protocol (MAP), Technical Office Protocol (TOP) , and the electric power industry) are cooperating to produce a common procurement specification for computer networking products and services based on the OSI international standards. This document is the *Industry/Government Open Systems Specification* (IGOSS) [13]. The IGOSS will enable the different user communities to speak to the vendors with one voice about their procurement and testing requirements. The IGOSS will result in a list of conformant products that is recognized by all participating organizations. Strong efforts will be made to align the IGOSS document with similar documents being produced in all regions of the world in order to maximize the degree to which interoperability can be achieved using nonproprietary solutions.

Version 3 of GOSIP will reference the IGOSS as the base document. The IGOSS, in turn, will reference the *OSE Implementors' Workshop (OIW) Stable Implementation Agreements* [25]. Part 14 of the OIW Stable Implementation Agreements specifies the agreements for Virtual Terminal. The IGOSS applies in the procurement of VT services provided by the following profiles:

1. the Generalized Telnet Profile,
2. the Forms Profile,
3. the Paged Profile, and
4. the X.3 Profile.

The Generalized Telnet Profile provides functionality identical to the TELNET protocol of the TCP/IP protocol suite. The Forms Profile supports forms-based applications with local entry validation of data performed by the terminal system. The Paged Profile provides forms capability typified by the existing block-mode terminals. The Paged Profile is specified by the International Standardized Profile (ISP) AVT23 [17]. The X.3 Profile provides functionality identical to the set of CCITT recommendations for PAD (X.3, X.28, X.29). For a more complete description of each of the above listed profiles, please refer to Section 4.

Version 2 of U.S. GOSIP, includes two categories of VT systems:

1. simple systems and

2. forms capable systems.

Simple systems support the Generalized Telnet Profile; forms capable systems support the Forms Profile. Version 2 of GOSIP [11] initially referred to the OIW Implementors' Agreements VT Profile, Telnet-1988. This version of the Telnet profile does not mandate support for any options, but limits the options which an implementation may support to BINARY, ECHO, and SUPPRESS-GOAHEAD. The Telnet-1988 Profile does not include the TCP/IP TELNET basic negotiation facility for expanded option support. The Generalized Telnet Profile which supports the above mentioned options and the negotiation facility for expanded option support provides a superior technical solution. For this reason, Version 2 of U.S. GOSIP has been modified to include the Generalized Telnet Profile instead of the Telnet-1988 Profile.

Chapter 6

Evaluation Guidelines

This chapter describes a process for selecting and evaluating candidate VT implementations. The functions a user should consider in selecting and evaluating candidate implementations are examined and data is provided on other factors which users may need to consider when evaluating candidate VT implementations.

6.1 Candidate Implementations

This section recommends a two-step procedure for creating a list of VT implementations, which are procurement candidates. First, the user creates a list of available VT implementations. The user may find available VT implementations by contacting computer and communications vendors, perusing computer and communications periodicals for product advertisements and announcements, consulting the list of GOSIP conformance tested products, and by attending computer and communications trade shows.

Second, the user defines any restrictions which apply to the candidate VT implementations. These restrictions may include specifying the hardware and operating system on which the candidate VT implementations must run, and specifying a price range for the candidate VT implementations. For example, if a user requires that candidate VT implementations run on an IBM PC-compatible system, only VT implementations running on an IBM PC-compatible system would be placed on the list of candidate implementations. After the user has determined the restrictions, the list of candidate implementations will comprise VT implementations which comply with all the user's restrictions. Once the list is created, product literature, users manuals, technical specifications, and other available information should be requested from each of the vendors for the candidate VT implementations. This information will provide input to the evaluation procedure.

6.2 Functional Evaluation Guidelines

There are several functions or features which need to be considered when procuring a VT implementation. This section provides users with valuable information on the functions or options available in VT implementations. Users should use the information to aid them in defining the list of restrictions which apply to candidate VT implementations.

This section describes functionality potentially available in a VT implementation. The functions presented here provide a representative sampling of the functionality currently available in VT implementations. The functions comprising this section were derived from the following sources:

1. the International Standard for OSI Basic Class Virtual Terminal Service and Protocol [14, 15],
2. the OIW Stable Implementors Agreements for Open Systems Interconnection Protocols [25], and
3. practical experience with, and review of documentation from, VT implementations [3, 4, 5, 6, 7, 8, 9, 20, 21, 28, 29, 30, 31] in the NIST Network Applications Laboratory.

The minimum functional requirements for a VT application are defined by the IGOSS; however, these requirements must be augmented to meet the needs of most users. The user should carefully study each function described to become familiar with the functionality potentially available in candidate VT implementations. Although many functions are presented here, it is not possible to include every function that is important to every user. The user may consider any functions which are not included here, but are important to that user.

6.2.1 Virtual Terminal Roles

This section assists users in determining their VT configuration. The VT configuration is useful in evaluating the functionality of VT implementations because it provides input to the user's functional requirements.

A VT configuration consists of two parts. The first part identifies the role of the VT implementation. A VT implementation may act in any of the following roles:

Initiator A VT implementation configured as an initiator can initiate VT requests to other VT implementations. For example, a user residing on an initiator-only configuration can initiate an association with a remote system or application.

Responder A VT implementation configured as a responder can respond to VT requests initiated by other VT implementations. For example, a remote user (i.e., a user not residing on the VT responder) can initiate an association with a VT implementation configured as a responder only.

Initiator and Responder A VT implementation configured as both an initiator and responder can initiate requests to remote VT implementations, and can respond to requests initiated by remote VT implementations.

Figure 6.1 depicts the different roles of a VT implementation. System A is initiator only, system B is responder only, and System C is both an initiator and responder system.

6.2.2 Network Type

The second component of the VT configuration is the network type. A VT implementation may be connected to a Wide Area Network (WAN) and/or a Local Area Network (LAN). The following examples describe the WAN, LAN, and Network Independent configurations:

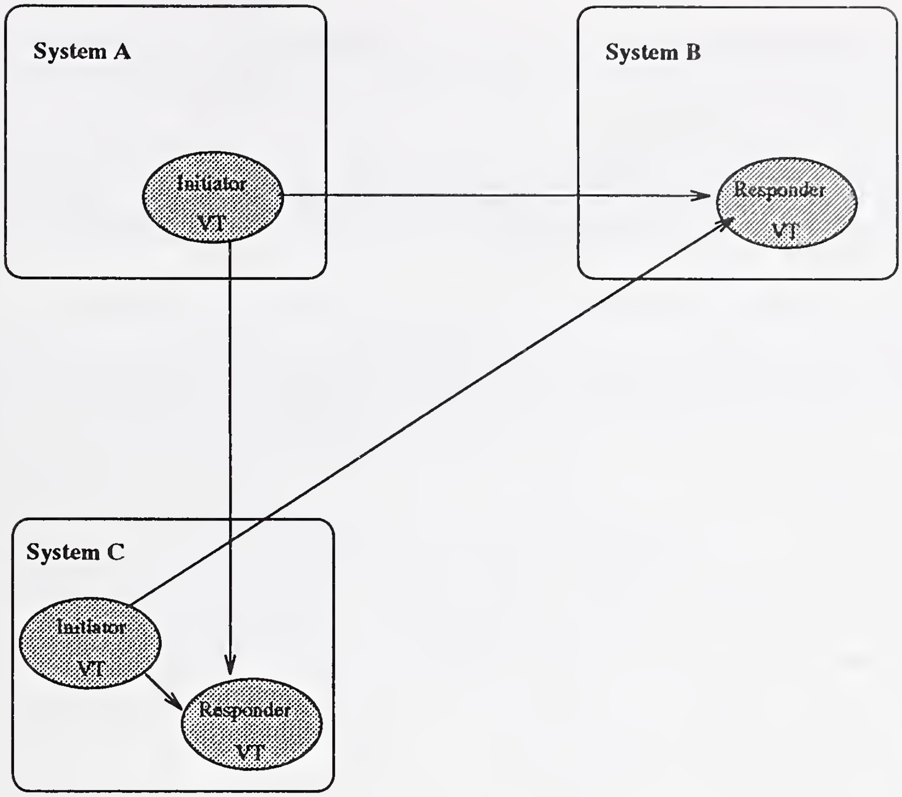

Figure 6.1: Initiator/Responder Relationship

Wide Area Network connection This configuration allows a VT implementation connected to a WAN to communicate with other VT implementations. For example, a company may have a computer system with a VT implementation connected to a Wide Area Network. Employees of this company may open associations with other companies having VT implementations that can access the same Wide Area Network.

Local Area Network connection This configuration allows a VT implementation connected by a LAN to communicate with other VT implementations. For example, a college campus may have computer systems with VT implementations located in different buildings on the campus; the computer systems are interconnected by a Local Area Network. Students may open associations between the computer systems that reside in the different campus buildings. It is also possible for a VT implementation on a LAN to access other VT implementations on different LANs across WANs using routers.

Network Independent connection This configuration provides for a number of different network solutions from which the user can pick and choose. This allows the user to buy only what is needed with the option to buy more later.

6.2.3 Functional Units

The VT service supports a number of capabilities called Functional Units. The functional units required are selected during VT association. The VT service defines five functional units; BREAK, Urgent Data, Switch Profile Negotiation, Multiple Interaction Negotiation, and Negotiated Release. The OIW Implementors' Agreements [25] only reference the BREAK, Urgent Data, and

Switch Profile Negotiation functional units. The following paragraphs describe the functional units specified in the OIW Implementors' Agreements and the conformance requirements for each.

The OIW Implementors' Agreements mandate support of the BREAK functional unit. The BREAK functional unit allows a VT user to destructively interrupt and discard the current sequence of display object and control object updates. This forces the resynchronization of the service provider

The Urgent Data and Switch Profile Negotiation functional units are specified as optional functional units in the OIW Implementors' Agreements. The OIW Implementors' Agreements define optional as "not required to meet minimum conformance requirements but identified as providing additional useful capabilities." The Urgent Data functional unit provides the capability to allow a small amount of data to be conveyed from one VT user to its peer in an urgent manner, possibly bypassing previous data exchanges. This functional unit is intended to be used for signalling high priority events. The Switch Profile Negotiation functional unit makes it possible for a user to change to a different profile without breaking off an association and renegotiating everything from the beginning. Please refer to section 3.4.4.1 for a more detailed description of Switch Profile Negotiation.

6.2.4 Implementation Profiles

One function or option to consider is vendor profile support. Many vendors support several profiles in a given product. This gives the user flexibility in applying the VT service to more terminal/application communication scenarios. Chapter 5 lists the profiles which provide the VT functionality specified by the Industry/Government Open System Specification. The OIW Implementors' Agreements [25] and the Virtual Terminal [14] standard require that a vendor support the default profile for the communication mode being implemented (A-Mode or S-Mode). The OIW Implementors' Agreements [25] further require a vendor to support at least one other profile in addition to the default for the communication mode selected. Chapter 4 provides a detailed description of the VT profiles specified in the Industry/Government Open Systems Specification.

6.2.5 Profile Option Support

One function to consider in selecting an implementation based on a profile that supports many options is which options of the profile are supported. An example of this can be found in the Generalized Telnet Profile. The most important differentiator will be the implementation's support of Telnet options, such as Binary Transmission and Status. It is through the Telnet options that specific user and procurement requirements will be met.

6.2.6 User Interface

The user interface of a product is another important function for a potential buyer to evaluate. For the most part there are no major differences in the user interfaces provided for VT products. A few ways in which one user interface may differ from another are presented here.

A user interface may provide a HELP facility. The capability to log the session may be supported. Some implementations may allow the user to change non-VT parameters like the command

prompt. Most interfaces support some means of returning to the operating system without terminating the association. Another difference is that some systems support list options for parameters related to the VT operation, like the current value of user-selectable parameters or a list of remote systems with which the user may establish associations. Some user interfaces may be specifically designed to be similar, if not identical, to a familiar user interface. For example, users migrating from TCP/IP to OSI would like to see the same TELNET interface they now use.

The above items do not represent an exhaustive list of differences for user interfaces, but are presented to give the user additional data to consider when preparing the list of requirements for candidate VT implementations. Information of this nature may be gleaned from reviewing product documentation or from product demonstrations.

6.2.7 Application Gateways

There are other protocols which provide the same functionality as OSI Virtual Terminal. One such protocol is the TCP/IP TELNET protocol. There are also many proprietary protocols in use today which provide functionality equivalent to OSI Virtual Terminal. One answer to allowing products based on other protocol solutions to work with OSI VT is application gateways. However, in some environments, gateways are not so important because the users themselves can act as gateways using the TCP/IP protocol suite to connect to one system then using the OSI protocol suite to connect to another.

An application gateway is an application written to interface between two or more products providing similar functionality. Such gateways aid in the migration from TCP/IP or proprietary protocols to OSI protocols. Application gateways already exist between OSI File Transfer, Access and Management (FTAM) and the TCP/IP File Transfer Protocol (FTP) and OSI Message Handling System (MHS) and the TCP/IP Simple Mail Transfer Protocol (SMTP). Application gateways between OSI VT Generalized Telnet and TCP/IP TELNET now appear in some released products.

6.2.8 Application Integration

One function which a user may require is integration of the VT application with other applications. This integration may take one of two forms:

1. integration with OSI applications and/or

2. integration with proprietary applications.

Directory Service (X.500) and Message Handling Systems (X.400) applications are the first OSI applications which are likely to be integrated with VT implementations. For example, a VT implementation can take advantage of a directory service implementation, by using the directory to look-up the address of a particular VT responder system.

Many vendors have proprietary products which they may integrate with their VT implementation. Users who have applications based on the proprietary products will find this an added, if not necessary, benefit. For example, the user may currently use a forms generating application from

Vendor A. If Vendor A releases a VT implementation which is integrated with its forms generating application, the user can use all the existing forms generated using Vendor A's proprietary product.

6.2.9 Administrative Functions

Administrative functions can be divided into three categories: administration, debugging, and access control. VT administration functions relate to the installation, configuration, and maintenance of the implementation. Section 6.2.9.1 describes possible VT administration functions. Debugging functions are useful for isolating and resolving problems. Please see section 6.2.9.2 for a description of VT debugging functions. Access control functions are used to limit access to the VT implementation. They are described in section 6.2.9.3.

6.2.9.1 Administration Functions

Virtual Terminal administrative tasks begin with the installation of the VT implementation. An implementation may provide an on-line training program for installing and configuring the implementation. The installation may be accomplished via an automated procedure which prompts the user for information. To ensure the installation was performed correctly, an installation verification facility may be provided.

Once installation is complete, the implementation must be started. The starting of the implementation may occur when the system supporting the implementation is started, or it may be started and stopped manually. Also, the implementation may be started as just an initiator or a responder. If the implementation is operating in both initiator and responder roles, one of the roles may be stopped without affecting the other role. For example, an implementation may be stopped from initiating VT commands, but can continue to respond to VT commands initiated by other VT implementations

Once the VT implementation is installed and operating, the two main administrative tasks are maintaining VT administration databases and optimizing the implementation. Virtual Terminal administration databases are typically used to store information pertaining to VT responders. This information includes underlying OSI layer addressing.

To assist with optimizing the implementation, VT statistics relating to implementation usage may be provided. Statistics such as the date and time the implementation was started, total number of bytes sent and received, average throughput, current user count, and current VT associations may be available. The statistics may be separated to show usage of the implementation in both initiator and responder roles. If the statistics show the VT implementation to be overloaded, specific parameters may be modified to optimize the implementation. An example parameter is the maximum number of simultaneous users supported by the implementation.

A VT implementation may provide a utility program to assist with performing administrative tasks. The utility program may be used to view and update VT database information and optimize VT parameters. The utility program may also contain an online help facility so that information such as commands recognized by the program, how they are used, and the options and parameters supported by the commands may be viewed.

A VT implementation may provide other administrative functions. A user may be able to backup the VT implementation, or restore the implementation from a backup. The backup may be restored to a different machine if the original machine is encountering hardware problems. Restoring an implementation is different from installing an implementation in that information registered in VT databases need not be re-entered when the implementation is restored.

6.2.9.2 Debug Capabilities

A VT implementation may provide functions which assist in resolving problems encountered. An implementation may provide a log file for VT activity (e.g., associations). Any errors encountered, such as those involving system resources, may also be logged. This information may be displayed on the system console in addition to being written to a file.

To aid in solving specific problems, an implementation may provide a utility that monitors and displays all VT protocol and data exchanges. For example, a user may be able to view the parameters and values transferred between two VT implementations during the establishment of the VT association. This utility may also perform tracing of the underlying OSI layers.

6.2.9.3 Access Control

A VT implementation, acting as a responder, may provide functions which limit the access of initiating VT users. Access control may be based on the initiating user's network address or initiator identity value. A VT implementation may deny access to specific network addresses or only allow access by specific network addresses. Similarly, access by certain initiator identity values may be denied or limited.

A VT responder may provide an account that may be used by any initiating user. This VT account, sometimes referred to as an anonymous account, is accessed by providing the correct initiator identity value (e.g., ANONYMOUS or GUEST) and any password value. The password is not validated. Since any user can access this account, the account is typically given minimal system privileges as a security precaution.

6.2.10 Underlying Open Systems Interconnection Layers

In addition to VT functions, a VT implementation may provide additional functionality in the underlying OSI layers. The ACSE Layer functionality determines the ACSE Layer implementations with which the VT implementation may interoperate. The OIW Implementors' Agreements [25] specify that only the Kernel ACSE functional unit must be supported. Presentation Layer functionality determines the Presentation Layer implementations with which the VT implementation can interoperate. The OIW Agreements [25] specify that only the Kernel Presentation functional unit must be supported. VT places definite additional requirements on the functionality of the Session layer. These requirements may vary depending upon the profile(s) supported. Session options are listed in part 5 clause 13.4 of the OIW Implementors' Agreements [25].

For IGOSS end systems, the connection-oriented transport service provided by Transport Class 4 is mandated for enterprise-wide interoperability. It is the required means for providing a reliable end-to-end communications path between end systems.

The Connectionless Network Service (CLNS) is also mandated for enterprise-wide interoperability. Together with Transport Class 4, it provides the means for interconnecting local and wide area subnetworks. The Connection-Oriented Network Service (CONS) is an additional, optional service that may be specified in conjunction with Transport Class 4 for communication among end systems that are directly connected to X.25 wide area networks and Integrated Services Digital Networks (ISDNs). Use of this service can, under certain circumstances, avoid the overhead associated with the CLNS. In addition, Transport Class 0 and Class 2 over the CONS may be specified to enable communication with systems that are not compliant with the Industry/Government Open Systems Specification.

If the VT implementation is to be used over an X.25 network, the X.25 software provided with the implementation should conform to the 1984 or 1988 version of the Consultative Committee on International Telephony and Telegraphy (CCITT) X.25 protocol. It may instead conform to the 1980 X.25 when 1984 or 1988 services are not available. To assist with setting X.25 parameter values, pre-defined groups of X.25 parameters specific to a network service may be included with the implementation. For example, if the VT implementation will be used over ACCUNET, the Wide Area Network provided by AT&T, the ACCUNET parameter group would provide all X.25 parameter values needed to establish X.25 ACCUNET connections. The provision of these values decreases installation time and optimizes X.25 performance. The X.25 parameters may be modified later to better suit any individual requirements.

6.2.11 Conformance and Interoperability Testing and Registration

Conformance testing, which verifies that an implementation conforms to the standard, is required by NIST in order for suppliers to claim legitimately to be GOSIP-compliant. Interoperability testing verifies that the implementation interoperates with other implementations. Interoperation testing with other implementations is optional, but recommended.

The NIST has defined a GOSIP Testing Program to permit federal agencies to substantiate claims of GOSIP compliance. If a supplier claims GOSIP compliance or conformance for a product, then a buying agency is advised to require that the product be tested in accordance with the criteria specified in the *GOSIP Conformance and Interoperation Testing and Registration* report [12]. If a product includes a multilayered GOSIP profile, then all protocols for which GOSIP compliance or conformance is claimed should be tested in accordance with these criteria. Federal agencies requiring claims of GOSIP conformance should consult the Register of Conformance Tested GOSIP products. Agencies that require that interoperability between GOSIP conformant products be documented should consult the data supplied by a service on the register of Interoperability Test and Registration Services.

Information such as which tests were performed, with whom, when, as well as the actual test results should be made available to the user. For more information on GOSIP testing, the user should read the *GOSIP Conformance and Interoperation Testing and Registration* document [12].

As of late 1992, there are no conformance test suites available for Virtual Terminal. The European Workshop for Open Systems (EWOS) has released a document titled, *Planning for Conformance Testing for VT* [23]. The EWOS document discusses the steps necessary to develop

a conformance test suite. It appears to concentrate on S-Mode VT service. No starting dates for this work are given in the EWOS document. It is difficult to say when there will be a conformance test suite for Virtual Terminal.

Where no conformance test suites exist, the GOSIP Testing program advises agencies to define their own acceptance criteria. One way of satisfying this criteria might be interoperability testing. Interoperability testing is not as comprehensive as conformance testing, but it does provide a level of confidence to the user that the product works and interoperates with other products based on the same protocol.

Development of a VT interoperability test suite is commencing under the direction of OSINET, an interoperability testing service that has been approved by the U.S. GOSIP Testing Program. The OSINET appears on the U.S. GOSIP register of interoperability testing services.

6.2.12 Hardware Requirements

Different VT implementations may require specific hardware for operation. VT implementations take one of two forms:

1. host implementations or

2. terminal servers.

Requirements pertaining to the CPU, disk space, memory, and external devices (e.g., X.25 interface cards) may need to be met. In addition, vendors may offer a family of hardware configurations (e.g., terminal servers with 4, 8, 16, or 32 ports).

6.2.13 Software Requirements

Different VT implementations may require specific software configurations for operation. For example, a VT implementation may consist of multiple software components which need to be installed separately. OSI software not residing in the Application Layer, such as lower layer software, may require installation. In addition, a VT implementation may need a specific operating system and version.

A function related to software requirements is vendor licensed source code. Vendors may license the source code to a VT implementation as part of the overall product. This feature may be important to the user who wishes to write a gateway to an existing in-house protocol.

6.2.14 Documentation

A VT implementation might provide a user with a variety of manuals explaining the interworkings of the application. Although each implementation might organize its documentation differently, the following information, regardless of format, may prove useful to the user: installation guide, user's guide, administration guide, troubleshooting guide, and quick reference guide. An installation guide provides information for installing and configuring the VT implementation. It may contain sample installations. A user's guide describes the VT commands recognized by the implementation.

An administration guide details management and maintenance of the VT implementation. A troubleshooting guide describes possible errors and how they are corrected. A quick reference guide, which is useful once the user is familiar with the implementation, provides a quick reference for VT commands.

6.3 Other Guidelines

This section describes other factors to consider when evaluating candidate VT implementations. The guidelines defined in this section are not as concrete as the ones in the previous sections, and therefore, are not in the functional evaluation section. They are, however, factors to be considered when evaluating implementations.

This section contains four major topics. The first topic, effectiveness, is relevant to the VT implementation. The second two topics, commitment and support, are relevant to the vendor. The final topic is cost.

One factor to consider when evaluating a VT implementation is the effectiveness of the functionality provided by the implementation. For example, a user may be provided with a program to assist with installing the implementation; however, the installation procedure may be very difficult and time-consuming despite the installation program. Debugging functions may exist, but may not facilitate the easy resolution of most problems. Finally, the documentation provided with an implementation may not be well organized, or may be difficult to understand.

To appreciate the effectiveness of a VT implementation, the user has several options. The user can request a copy of the VT documentation from the vendor. By examining the documentation in advance, the user can better determine its adequacy and understandability. The user may also be able to determine how easy the implementation is to install, configure, debug, and use. Another option is for the user to request a demonstration of the VT implementation. A demonstration will provide an overall view of the implementation, especially concerning its "user friendliness."

Some evaluation factors relate to the vendor. The user should consider the company's commitment to OSI, and if the personal contacts (i.e., sales and service representatives) are well informed. The user may consider whether the company marketing the product also developed the product. Other noteworthy evaluation factors are the company's ability to service their product, the company's policy regarding product upgrades, and customer service issues. Customer service issues include: software support, whether the support is local or out of town, and maintenance agreements. The user should ask the vendor about the type and extent of customer support that is available.

The final topic concerning the evaluation of a VT implementation is the cost of the implementation. This includes hardware costs (e.g., computer systems, LAN cards, WAN cards), software costs, and maintenance costs, such as maintenance contracts. The budget of the user will determine the importance of cost as an evaluation factor.

Chapter 7

Virtual Terminal Products

This chapter provides the user with a forecast of the VT market both near and long term. The OIW Implementors' Agreements [25] are complete for all but one VT profile. The Forms and X.3 profiles were completed in 1989. The Generalized Telnet Profile was stabilized in December of 1991. Implementors' Agreements for the Paged Profile are still in progress and this work is being handled at the international level through the International Standardized Profile (ISP) process.

Specifications, such as GOSIP and the IGOSS, which contain VT functionality encourage the development of VT products. As the reader will recall from Chapter 5, four VT profiles are specified in the IGOSS:

1. Generalized Telnet Profile
2. Forms Profile
3. Paged Profile
4. X.3 Profile

In the past, extensive product availability coincided with the date GOSIP became mandated, i.e., a sufficient number of applications based on protocols specified in GOSIP were available from vendors about the time that that version of GOSIP became mandatory in Federal procurements. It is expected that the same pattern will hold for Virtual Terminal Applications, since the above profiles are specified in IGOSS [13], which is referenced by Version 3 of the Government Open Systems Interconnection Profile.

Vendors attending the OIW VT Special Interest Group (SIG) were polled to provide a more exact picture of the VT market. Each participant was asked to assess the VT market using the following time periods:

- today
- within 1 year
- within 2 years
- within 4 years

The following paragraphs summarize their responses.

Today, users can procure VT implementations supporting several A-Mode profiles. One such profile is Telnet-1988 Profile. Several vendors have implemented the Telnet-1988 Profile. Telnet-1988 Profile implementations are available primarily because the profile was originally specified in GOSIP Version 2. Vendors who implemented the Telnet-1988 Profile are likely to include it as part of an A-Mode VT implementation. Vendors who market in Europe offer the Telnet-1988 Profile, since it satisfies a U.K. GOSIP [27] requirement.

In addition to the Telnet-1988 Profile, today's VT implementations support the A-Mode Default and Transparent profiles. As the reader may recall, it is an OIW requirement that A-Mode implementors support the default profile. The Transparent Profile is generally supported by any A-Mode implementor. Please see Section 4.4 for a description of the Transparent Profile.

In the coming year, users will increasingly see vendors supporting the Generalized Telnet Profile. There are several reasons for this. First, it is the Generalized Telnet Profile which replaces the Telnet-1988 Profile in the GOSIP Version 2 specification. Second, the IGOSS specifies the Generalized Telnet Profile as a valid VT profile. Finally, the Generalized Telnet Profile is a direct competitor with and replacement for TCP/IP TELNET. Vendors who are strong supporters of OSI will no doubt include the Generalized Telnet Profile and a Generalized Telnet/TELNET gateway in their VT products to aid in migration to OSI Virtual Terminal. It should be noted that the Generalized Telnet Profile does not interoperate with the Telnet-1988 Profile.

Vendors responded that in two years VT implementations will include the following additional profiles:

- Forms Profile
- Paged Profile
- X.3 Profile
- possibly, Scroll Profile

Once again the appearance of the above mentioned profiles is directly related to the VT profiles specified in the Industry/Government Open System Specification [13]. Forms Profile implementations may appear slightly ahead of the other profiles, because the Forms profile is specified in GOSIP Version 2 [11]. One vendor listed the Scroll Profile with the caveat that it will be implemented in this time period if agreements for this profile become stable.

The Scroll Profile provides line-at-a-time interactions between a terminal and host system, allowing bi-directional (forward and backward) scrolling of the display. The profile includes the ability to switch local echo on or off and supports the "type-ahead" ability. A typical use for this profile is for applications where type-ahead may be advantageous and control over local echo "on"/"off" is required, e.g., the type of application where a conventional teletypewriter device or "teletype-compatible" video device having "full-duplex" capability is often used. Implementors' agreements for the Scroll profile are in progress but are not stable at this time.

Additional VT implementations supporting the Generalized Telnet Profile may appear on the market. Vendors may upgrade their Generalized Telnet Profile implementations by including additional options which have been defined for TCP/IP TELNET. One attraction of the Generalized Telnet profile is that it has been specifically designed to allow future options of TCP/IP TELNET to be automatically incorporated into the Generalized Telnet Profile.

In 4 years, users may see a few more X.3 Profile and Forms Profile applications. Vendors were reluctant to speculate further. Most vendors polled felt that implementation of additional profiles depended on the market. If anything, users will find that there are more vendors marketing the already-mentioned profiles, rather than a small number of vendors continuing to expand their VT implementations to include new profiles.

Appendix A

Virtual Terminal and X Windows

This appendix discusses the X Window system and the integration of the X Window system over Open Systems Interconnection. A brief history of OSI VT is given to help the reader understand how X Windows may provide a solution to an obvious problem. A brief description and history of the X Window system is provided, as well as a description of the mapping of X Windows over Open Systems Interconnection. Finally, the current status of X Windows is given.

The only interactive terminal protocol addressed by standards bodies for OSI is the ISO Virtual Terminal service and protocol [14, 15]. Originally, the VT standard envisioned supporting five classes of terminal-related applications, one of which was a graphics/windows class. As international participation for VT diminished, work on classes beyond the Basic Class was never introduced. Thus, the OSI VT standard is insufficient to meet the requirements for today's windowing applications.

Today's windowing implementations are based on the X Windows X11 specification. The X Window protocol provides a multilevel programming environment, a display manager, an application protocol, a structured library for building user interfaces, and a collection of Client application programs. Simply stated, the X11 is a de facto standard for distributed window management. The X11 was designed and developed via an iterative process at M.I.T. and by third party contributors. The X Consortium, a large and influential collection of hardware and software vendors, currently supports maintenance and development of the X11 specification.

The X11 specification is seen by many as the answer to the OSI VT windowing application dilemma. Subcommittees 21 (OSI) and 24 (Graphics) of the ISO/International Electrotechnical Committee (IEC) Accredited Joint Technical Committee (JTC1) believe the widely used X11 Release 3 and Release 4 Window Systems [24] are first generation window systems that should be adopted for use on systems implementing OSI standards.

Early in 1989, several proposals were submitted to the American National Standards Institute (ANSI) Accredited Standards Committee X3H3.6 to consider support of X11 in Open Systems Interconnection. The X3H3 committee selected an Application layer mapping which uses a minimal set of primitives and functionality from existing OSI Application layer standards such as the ACSE and the Presentation layer. This mapping places X11 at the Application layer. The ACSE handles

association establishment and termination, with the establishment service primitives mapped to the X11 open display function and the termination service primitives mapped to the X11 close display function. The mapping supports an abrupt close option, for whatever reason, on the Client side. All other X11 operations make use of the service primitives of the Presentation layer. Figure A.1 shows the mapping of X11 over the ACSE and Presentation layers.

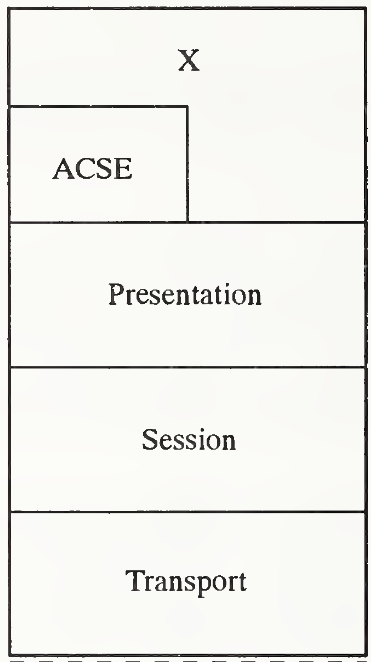

Figure A.1: X Windows to OSI Mapping

Like many network based applications, X11 is based on a Client/Server model, which allows applications and resources to be distributed and shared across a network. The X11 Client/Server model differs from the typical definition of a Client/Server architecture. A Client in X11 is not associated with a human user, but rather a potentially remote application program that needs to display information on one or more workstations. A number of Clients share the resources and display services provided by the workstations. The X11 Server is responsible for controlling the display, as well as passing information between the workstation user and the X11 Clients. The Server's responsibilities include such items as displaying Client output, acting on Client requests, issuing events and error messages, and passing all user input from devices like a keyboard or mouse to the appropriate Clients. Figure A.2 shows the X11 Client/Server relationship.

The proposed ANSI standard is now a four-part standard: parts I - III provide the X Windows data stream definition, part IV defines the mapping of X Windows data streams onto OSI ACSEs and the Presentation service. The VT SIG of the OIW has been following the development of the proposed X Windows mapping over OSI and concludes that no implementation agreements will be required for the mapping. The OIW VT SIG modified their charter to handle certain X Window related tasks, if necessary. These tasks include monitoring the X Windows system and potentially developing implementors' agreements, registering and maintaining OIW X-OSI objects and reviewing and generating abstract test cases for X Windows. The only issue that must be addressed is the means of initiating remote X Clients.

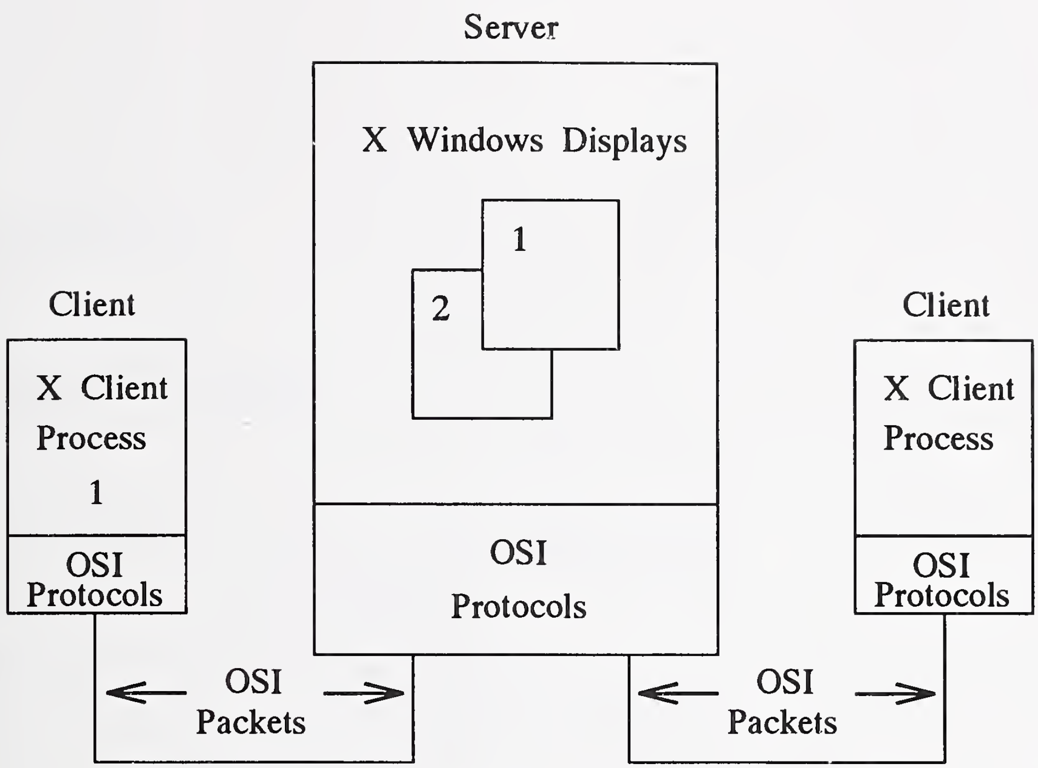

Figure A.2: X11 Client/Server Relationship

The IGOSS includes X Windows applications operating over an OSI-based network. The IGOSS supports X Client or X Server applications or a combination of both. The IGOSS specifies use of the VT service as the means to initiate remote X Clients. It is expected that user demand will result in products implemented according to the ANSI standard even as the X11 standard is "fast-tracked" through the ISO standardization process.

Appendix B

Abbreviations

This appendix defines the abbreviations used in this document.

ACSE Application Control Service Element

ANSI American National Standards Institute

ASCII American Standard Code for Information Interchange

CCA Conceptual Communications Area

CCITT Consultative Committee on International Telephony and Telegraph

CLNS Connectionless Network Service

CONS Connection-Oriented Network Service

CR Carriage Return

EG Expert Group

EWOS European Workshop for Open Systems

FEPCO Field Entry Pilot Control Object

FEICO Field Entry Instruction Control Object

FIPS Federal Information Processing Standard

FTAM File Transfer, Access, and Management

FTP File Transfer Protocol

GOSIP Government OSI Profile

IEC International Electrotechnical Committee

IGOSS Industry/Government Open Systems Specification

IP Internet Protocol

IRV International Reference Version

ISDN Integrated Services Digital Network

ISO International Organization for Standardization

ISP International Standardized Profile

JTC1 Joint Technical Committee 1

LAN Local Area Network

LED Light Emitting Diode

LF Line Feed

MAP Manufacturing Automation Protocol

MHS Message Handling Systems

MIN Multiple Interaction Negotiation

NIST National Institute of Standards and Technology

OIW OSE Implementors' Workshop

OSE Open Systems Environment

OSI Open Systems Interconnection

PAD Packet Assembler/Disassembler

PDISP Proposed Draft International Standardized Profile

POSIX Portable Operating System Interface

SIG Special Interest Group

SP Special Publication

SMTP Simple Mail Transfer Protocol

TCP Transmission Control Protocol

TOP Technical Office Protocol

TR Technical Report

VT Virtual Terminal

VTE Virtual Terminal Environment

VTP Virtual Terminal Protocol

WAN Wide Area Network

Appendix C

Glossary

This appendix provides a glossary of VT terms.

access right - An abstract object that confers on the owner, a VT user entity, the right to perform any of a particular set of operations that are defined as being controlled by the access right. An access right may be reassignable, (i.e., ownership of it may be transferred from one owning VT user to the other, previously nonowning, VT user), or nonreassignable, (i.e., one VT user owns it for the duration of its existence).

A-Mode or asynchronous mode - A mode of operation utilizing two monologues (in opposing directions) to provide update access to two display objects , one monologue for each display object; update access to each display object is controlled by a different, nonreassignable access right.

control object - Control objects are abstract objects which can be used to model aspects of real terminals such as bells, LEDs, or to handle control information.

current VTE - The single VTE that exists for a VT association when negotiation is not in progress.

device object - Device objects are used to represent real devices such as the keyboard, the screen, the printer, or a lightpen.

display object - A display object is an abstract object, used to model devices such as the screen or keyboard. Display objects are defined and may be visualized as one, two, or three-dimensional arrays of character-box elements.

draft VTE - The VTE, if any, that is under negotiation. During negotiation the draft VTE is not necessarily a complete Virtual Terminal Environment.

full VTE - A VTE that is a complete VTE, i.e., all parameters have been given values.

initiator - The process which serves the user who initiated the communication flow.

net-effecting - The conversion of a sequence of items, representing the content of one or more update operations into a different, usually shorter sequence, which results in the same final states of the objects being updated.

profile - A profile is a subset of features of the VT protocol that must be supported for a given connection.

responder - The process on the remote side which provides the services required by the initiator.

S-Mode or synchronous mode - A mode of operation utilizing two-way alternate dialogue supporting one display object; update access to the display object is controlled by a single, reassignable access right.

virtual terminal - A virtual terminal models the operations which may be performed on a terminal, and which are understood by both terminal and application.

virtual terminal association - An application association between two peer VT users.

virtual terminal environment - The context within which the virtual terminal will reside is known as the virtual terminal environment

VTE parameter - An individual parameter of a Virtual Terminal Environment.

VT user - A user of the Virtual Terminal Service.

References

[1] Boland, T., Government Open Systems Interconnection Profile Users' Guide, Version 2, Natl. Inst. Stand. Technol. Spec. Publ. 500-192, October 1991.

[2] Brennan, R., Thompson, K., and Wilder, R., Mapping the X Window onto Open Systems Interconnection Standards, IEEE Network Magazine, May 1991, pp. 32-40.

[3] Communications Server - Reference Guide - OSI/TCP, June 1991, 3COM.

[4] DECnet/OSI for ULTRIX - Installation, January 1992, Digital Equipment Corporation.

[5] DECnet/OSI for ULTRIX - Installation Roadmap, 1992, Digital Equipment Corporation.

[6] DECnet/OSI for ULTRIX - Introduction and Planning, November 1991, Digital Equipment Corporation.

[7] DECnet/OSI for ULTRIX - Release Notes, January 1992, Digital Equipment Corporation.

[8] DECnet/OSI for ULTRIX - Virtual Terminal Use and Management, Digital Equipment Corporation.

[9] EP/IX VT Telnet/4000 Package Release Notes, February 1992, Control Data Corporation.

[10] Government Open Systems Interconnection Profile (GOSIP), Version 1. Natl. Inst. Stand. Technol. Fed. Info. Process. Stand. Publ. (FIPS PUB) 146.

[11] Government Open Systems Interconnection Profile (GOSIP), Version 2. Natl. Inst. Stand. Technol. Fed. Info. Process. Stand. Publ. (FIPS PUB) 146-1.

[12] GOSIP Conformance and Interoperation Testing and Registration, Natl. Inst. Stand. Technol. NISTIR 4594.

[13] Industry/Government Open Systems Specification - Version 1 Draft, October 1992.

[14] Information Processing Systems - Open Systems Interconnection - Virtual Terminal Service - Basic Class, ISO 9040.

[15] Information Processing Systems - Open Systems Interconnection - Virtual Terminal Protocol - Basic Class, ISO 9041.

[16] Information Technology - Framework and Taxonomy of International Standardized Profiles - Part 2: Taxonomy, ISO/IEC TR 10000-2, July 1991.

[17] Information Technology - Internationalized Profile AVT2nn - Virtual Terminal Basic Class - Register of Application Profiles AVT1n, AVT2n, Part 2: AVT23 - S-Mode Paged Profile Third Draft PDISP 11187-2: 1991, December 1991.

[18] Markovitz, P., Trus, S., and Royster, C., Guidelines for the Evaluation of File Transfer, Access and Management Implementations, Natl. Inst. Stand. Technol. Spec. Publ. 500-196, October 1991.

[19] McLeod-Reisig, S., and Huber, K., ISO Virtual Terminal Protocol and its Relationship to MIL-STD TELNET. The Computer Society of the IEEE Technical Committee on Computer Committee Proceeding, November 1986, pages 110-119.

[20] OSI/ES with Demand Protocol Architecture, April 1991, 3COM.

[21] OSI/ES Internet BIOS Guide, 3COM.

[22] Perera, R., Virtual Terminal Tutorial, Retix.

[23] Planning for Conformance Testing for VT, Project Report - Part 1, EWOS/EG VT/91/85, August 1991.

[24] Scheiffler, R. W., X Window System C Library and Protocol Reference, M.I.T. and Digital Equipment Corporation, 1988.

[25] Stable Implementation Agreements for Open Systems Interconnection Protocols Version 5, Edition 1, Natl. Inst. Stand. Technol. Spec. Publ. 500-202, December 1991.

[26] Trus, S., Royster, C., and Markovitz, P., Guidelines for the Evaluation of Message Handling Systems Implementations, Natl. Inst. Stand. Technol. Spec. Publ. 500-182, August 1990.

[27] UK Government OSI Profile, 4th Edition, May 1991.

[28] Using VTP for AViiON Systems, Data General, February 1991.

[29] VT-720 Administrator Guide, June 1991, Retix.

[30] VT-720 for Interactive UNIX System V/386, Release 2.0.2, Installation Instructions, May 1990.

[31] VT-720 User Guide, June 1991, Retix.

NIST-114A (REV. 3-89)	U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY	1. PUBLICATION OR REPORT NUMBER NIST/SP-500/205
		2. PERFORMING ORGANIZATION REPORT NUMBER
	BIBLIOGRAPHIC DATA SHEET	3. PUBLICATION DATE November 1992

4. TITLE AND SUBTITLE

Guidelines for the Evaluation of Virtual Terminal Implementations

5. AUTHOR(S)

Carol A. Edgar

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

National Institute of Standards and Technology, Gaithersburg, MD 20899
Internal Revenue Service, 2500 Wilson Blvd, Suite 402, Arlington, VA 22201

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document advances the goals of the Government Open Systems Interconnection Profile (GOSIP) by providing guidelines for evaluating Virtual Terminal (VT) implementations. These guidelines can assist a user in the determination of which implementation, among several candidates, will best meet the functional requirements of that user. Specifically, this document contains: (1) guidelines for evaluating the functional specifications of VT implementations and (2) guidelines for matching the functional specifications of a VT implementation to the functional requirements of the user.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

functional evaluation; Government Open Systems Interconnection Profile; Industry/Government Open Systems Specification; initiator; Open Systems Interconnection; profiles; responder; TELNET; Virtual Terminal

13. AVAILABILITY

[X] UNLIMITED
[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
[X] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES

47

15. PRICE

ELECTRONIC FORM

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name ____

Company ____

Address ____

City ________________ State ____________ Zip Code __________

(Notification key N-503)

*U.S. G.P.O.:1992-341-832:60378

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bi-monthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.